Vulnerabilities, Influences and Interaction Paths: Failure Data for Integrated System Risk Analysis

Jane T. Malin
NASA Johnson Space Center
2101 NASA Road 1
Houston, TX 77058
281 483-2046
jane.t.malin@nasa.gov

Land Fleming
MEI Technologies, Inc.
2525 Bay Area Boulevard
Houston, TX 77058
281 483-2055
land.d.fleming1@jsc.nasa.gov

Abstract—[1,2]We describe analysis methods for identifying and analyzing cross-subsystem interaction risks from subsystem connectivity information. These methods identify entities that can pose a hazard to a function if they can be propagated from the hazard source to the vulnerable function. The analysis method can assess combined impacts of multiple disabling influences on a vulnerable function. The analysis method also uses numerical estimates of hazard strength to calculate cumulative measures of impact severity. These methods can support design of more dependable and diagnosable systems and promote better communication among subsystem designers about risk. Identification of cross-subsystem hazard-vulnerability pairs and propagation paths will increase coverage of risk analysis and can help identify risk control and protection strategies.

TABLE OF CONTENTS

1. INTRODUCTION

Many types of hazard and risk analyses are used during the life cycle of complex systems [16], including Hazard Analysis [13], HAZOP [19], Failure Modes, Effects and Criticality Analysis (FMECA) methods [14], Fault Tree and Event Tree Analysis, Reliability Block Diagrams, Layer of Protection Analysis (LOPA) [2], Hazard Indexes [15], accident analysis [5] and Probabilistic Risk Assessment, among others. The success of these primarily manual methods is dependent on the availability of input data and the knowledge that is applied by the analysts. It is common for the same organization to use more than one method as the life cycle progresses, and for an analysis team to perform the analysis without the benefit of the inputs or outputs of the work of the other teams.

We envision model-based and library-based automation that will assist in achieving reusable and repeatable analyses and analysis products. We have two main objectives: 1) better design of dependable systems based on better integrated cross-subsystem risk analysis and 2) reusable data that provides a basis for designing systems for diagnosis and impact analysis later in the life cycle.

Risk analyses are needed early in system concept development, at a system and subsystem level, so that designers can identify and take account of interactions between subsystems that can cause failures [6]. There have been several efforts to develop analyses that can be used in early stages in system acquisition, when there may be a functional design or conceptual designs that includes allocation of functions to subsystem architectures. These include methods based on functional requirements: Functional Hazard Assessment [1, 20], Function-Failure Design Method [14] and our work on Functional Failure Modes and Effects Analysis (FMEA) and Model Based Hazard Analysis [8, 9, 18].

Use of standard nomenclature, ontologies and libraries can make risk analyses reusable and maintainable, as well as less time-consuming and more complete and focused. Stone et al. [14] are developing empirically based libraries and nomenclature for mechanical systems. We have been developing spacecraft ontology and libraries of types of subsystems, functions, entities, hazards and failures for hazard analysis [11]. The ontology is broad and general, covering hardware, software and human systems. This nomenclature helps extract system connectivity (interaction) models from requirements text, by matching equivalent names and specializations of more general terms [17].

Our approach to better cross-subsystem risk analysis is intended to find more of the "unexpected" problems that are missed because complex integrated systems are difficult to analyze manually. Problems can be missed if they are

1——————————————

[1] 0-7803-9546-8/06/$20.00© 2006 IEEE
[2] IEEEAC paper #1450 Revision 2, updated December 13, 2005

dependent on an unusual or unexpected configuration of the system. We use abstract system connectivity models that can include external factors and interfaces among subsystems. These models are designed to be simpler and easier to maintain than risk models. System states and component modes can be reconfigured to support finding configuration-dependent problems and analyzing effects of risk controls. We derive potential source-target interactions and risk severity information from them.

Cross-subsystem interactions typically transmit "remote" hazards, which are not immediate and local hazards to vulnerable entities. We use hazard and vulnerability information to pair up an "unexpected" remote hazard source with a target entity that is vulnerable to the hazard. This can be a useful starting point for discussion of subsystem interactions between development teams, even without further information about propagation paths.

Logical trees of states or events are typically developed to model the causal relationships between hazard sources and impacted targets. To design systems for diagnosis and impact analysis, it is useful to analyze the combined effects of threats that by themselves may be only minor influences on failures and mishaps, but collectively may pose a severe hazard. This requires more than purely logical (Boolean) representations of hazard causes and effects [4]. The NASA Preliminary Hazard Analysis [13] process also calls for the analysis of failure propagation. For each vulnerable component in a system connectivity model, our method identifies hazard sources that match the vulnerability. A search is then conducted for paths between source and target. The approach uses physical system connectivity models to search for paths, taking account of system configuration (states of components and connections). A forward-simulation propagation analysis method has been developed for the HAZOP domain [19].

If a path is found from a hazard source to the vulnerable component, a numerical estimate of the impact of the hazard is calculated. There are many methods of ranking or prioritizing risks. A common qualitative ranking method uses a risk or criticality matrix with ratings of hazard likelihood and impact severity. Probabilistic Risk Assessment provides a quantitative method that requires detailed likelihood estimates. We calculate threat severity from two estimates: 1) worst case severity of threat influences on a vulnerable function and 2) threat propagation effectiveness of connections in a configuration.

Some methods assess risk controls, protections, counteractions and mitigations. Recommendations are developed for design changes or new functional requirements [1, 7, 13]. Few methods provide a way of validating or evaluating such controls, apart from ranking or prioritizing them, based on cost and effectiveness [3]. We can extend our approach to search for potential areas of control in a system and evaluate the effectiveness of proposed controls in system configurations. Our method captures the combined influences of hazards and propagation paths that would concern designers of controls and protections.

Impacted targets can be described as vulnerable to hazardous entities (e.g., materials, signals) that may be produced elsewhere in the system. A vulnerability is a sensitivity to local inputs that can contribute to losses or decreases in reliability (e.g., vulnerability to abrasive materials that may cause wear and early failure). A hazard has a negative impact on a function, even though it may be a legitimate product or by-product of a subsystem function. For example, Freon may be necessary resource for one process but could be detrimental to the functioning or physical integrity of another. In our models, the processes, behaviors or actions of components are not inherently functional [10, 18]. There is no distinction between a functional effect and a side effect except by explicit declaration. Likewise, effects need not be inherently hazardous, but can depend on system configuration.

In our initial approach, the scope of vulnerabilities is limited to functions and the scope of hazards is limited to entities. We plan to extend the scope to integrity of vulnerable entities (not just functions) and states of hazardous entity (not just entities regardless of state).

In this paper, we first describe a spacecraft case to illustrate the models and information used for analysis. Next we describe the model analysis capabilities:

(1) A path-finding procedure that identifies pairs of components in a connectivity model. An entity produced by a source component is considered to be hazardous if a path is found to a component function that is vulnerable to that disabling entity

(2) A scheme for estimating the effectiveness of a Source component at producing hazards and the degree of vulnerability of other components to hazards

(3) A hazard impact analysis method using an And-Or tree of disablers and a search for the path with the maximum degree of hazard transmission effectiveness. This method accounts for the transmission effectiveness of components on the path.

Finally, we describe our concept of knowledge acquisition, conclusions and concepts for future work.

2. Example Case

Figure 1 shows a simple component-connection model of a spacecraft. The "components" here are complex subsystems. The model was constructed in our Hazard Identification Tool (HIT), built on Protégé, a knowledge-base development tool. The model structure and functions were extracted semi-automatically by parsing a requirements specification, using a standard nomenclature, or ontology [12]. Then the model was manually elaborated to focus on a redundancy strategy for mitigating a noise problem. The model has four kinds of component connections: power (pwr), thermal (heat), and connections of various subclasses of the Data class. The system has a primary radio transmitter, labeled SC1_xmitter1 and a backup, labeled SC1_xmitter2. We can use this model to find components (or subsystems) that are vulnerable to entities generated by other subsystems.

Several kinds of information are needed for our analysis approach. This information can be supplied by the users or derived from documents, requirements tools or libraries. The kinds of information used for the analyses are:

(1) Components – subsystems or parts that can have state, operating modes and behavior, and can be connected to other components that can provide inputs or receive outputs

(2) Functions, processes, or behaviors that a Component can exhibit, called "Actions"

(3) Entities (e.g., fluids, power, heat, noise) generated or transmitted by Actions (of subclass "Output") of Components, which may be hazardous

(4) Vulnerabilities of component Actions that may be Disabled or degraded by exposure to Entities of various types

(5) Types of physical connections (e.g., solid, liquid, gas) across which an Entity of a given class may be propagated between Components

(6) Configuration – component states and Modes that may be the context for an Action or may control whether a potential hazard can reach or can be prevented from reaching a vulnerable entity

These data types are described in more detail with reference to the example model. The thermal control system (SC1_ThermalSys) Component outputs electrical noise (Output: Noise Entity) when it is operating (Action: Supply Heat). The transmitters are vulnerable to electrical noise (Disabler: Noise Entity) when transmitting (Mode: Transmit; Action: Transmit Data). When either transmitter is operating, electrical noise can interfere with the transmission of data from spacecraft to ground. The pwr connections can transmit power and noise entities.

HIT path analysis discovers a path over which this electrical noise from the thermal control system can reach the transmitters. This hazard/vulnerability pair that crosses subsystems may go unnoticed by the design groups. The design group responsible for the thermal control system would probably be aware that some components generate electrical noise while the design group responsible for the transmitters would be aware that electrical noise poses a hazard to data transmission. But neither group may be aware that a path exists for noise generated by the thermal system to reach the transmitters.

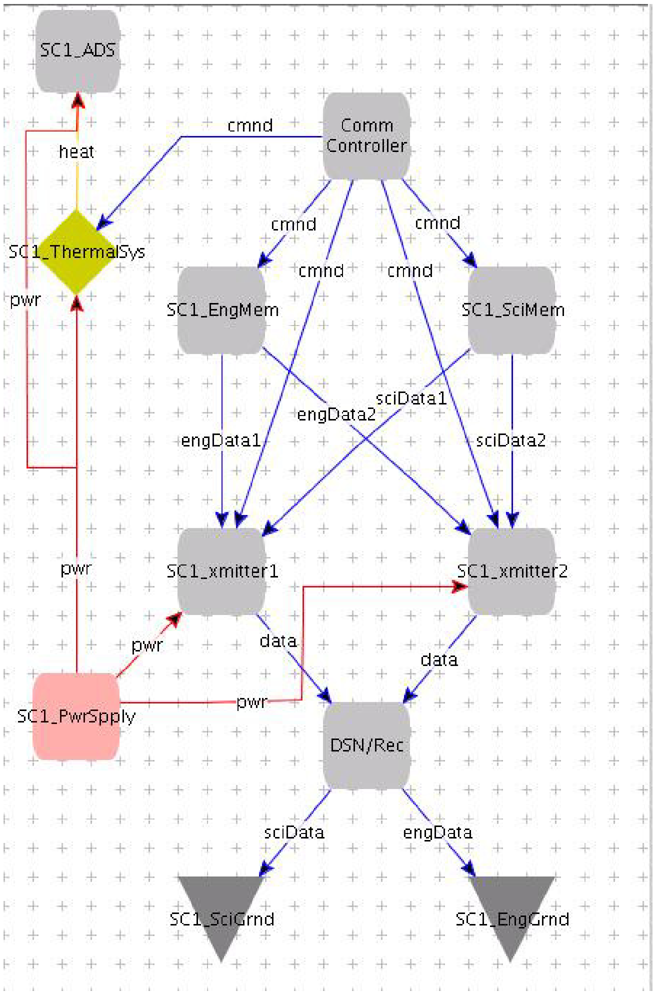

Figure 1 – Model of interacting spacecraft systems.

Integrated evaluation of degree of impact can be derived from this structural information. We will discuss impact estimation in Section 6. This assessment uses local estimates of: 1) Action Effectiveness (producing Hazardous Entities); 2) "Conductivity" of Connections and Actions and 3) Degree of vulnerability of Functions.

3. Searching for Pairs and Paths

Component Modes and Actions

By associating functions and side effects with component modes in a system model, hazard scenarios can be defined that are constrained by configuration. For path analysis, functions and side effects are allocated to operating modes of components. For example, a pump component may have the operational modes ON and OFF. The ON mode has the functional action "pumping fluid", a side effect action of "heating fluid" due to friction and possibly a second side effect of "generating contaminants" that enter the fluid. In the OFF mode, the function and side effects would not occur. Generating contaminants is viewed as a hazardous side effect of the ON mode rather than as a failure mode.

Figure 2 illustrates the relationships among functions, actions and modes of components. A component has a set of operational modes, one of which must be the component's current mode. Multiple actions (and their effects) can occur in a mode. The functions of a component are associated with specific Actions. The Boolean label "functional?" is used to select the actions that are functional. Side-effects are given the "functional?" value False. While in general there is a many-to-many relationship between modes and actions, in most cases an action will be associated with only one component mode.

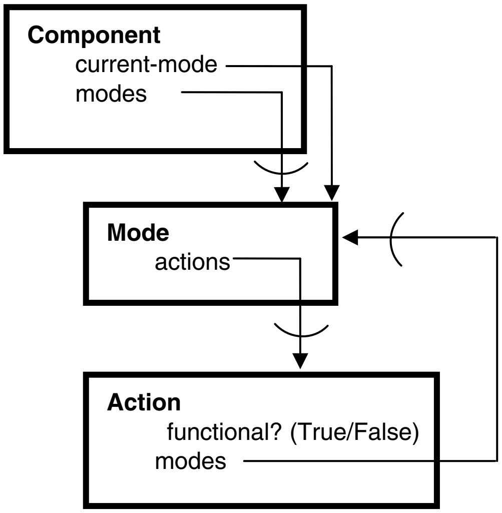

Figure 2 - Relations among Component, Mode, Action and Function. Object class names are **Bold**; attributes are indented; arrows represent "has-part" relations and arcs indicate they are multi-valued.

Component Connections

Relationships among actions, propagated entities, and types of component connections in a system model can also constrain hazard propagation. Actions produce entities that may propagate via component connections. In a model configuration in which component connections are being traversed, all components on the path from the source Component to a potentially vulnerable Component must also manifest a Conduction Action that can propagate the entity. As illustrated in Figure 3, component connections are treated as objects because the connection type of the object is important to guiding the path traversal procedure. At each step, the procedure checks the list in the Entity's "carrier-types" attribute. The Connection object can propagate the Entity only if the connection's type matches a carrier-type class in the list.

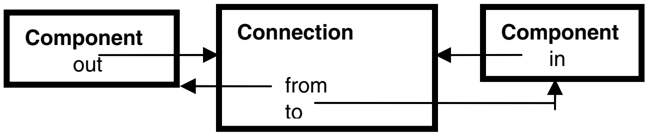

Figure 3 - Relations between Components and Connections

Vulnerable Functions are "Disabled" by Hazards

Functions can be disabled by actions of Sources that output entities to which the function is vulnerable. Entities that are associated with actions can be constrained by configuration and control. The search proceeds top-down from Target functions to hazard Source Actions.

The hazard specification scheme is illustrated by the diagram of Figure 4. An Action disabler represents a vulnerability of a function. Each instance, *A*, of a functional Action in a model has a "disablers" attribute that describes types of source actions performed by other components that will output Operand Entities that "disable" the Action. An Operand Entity is the entity that is output by an action. If Action *A* is vulnerable to a type of source Action, the source Action is represented by an action exemplar instance, *ExS*, in the "disablers" attribute of *A*. The Operand Entity output by Source Action *ExS* indicates the nature of the hazard. Attributes of the entity output by *ExS* will be used in the future to describe states that make the entity hazardous to an Action. Currently, the entity is assumed to be inherently hazardous. The Exemplars are matched to actual instances of Actions manifested in the system's current configuration during path search.

Disabler trees can be constructed to represent vulnerability to multiple disablers. If the output of more than one kind of source Action can disable a target Action, this is specified by setting its "disablers" attribute to an Or-Relation. The Or-Relation has an "action-group" attribute that points to either a set of Source Action exemplars, a set of And-Relations, or a combination of both. If a group of Actions must all be manifested in the system before the Action is disabled, the "disablers" attribute would be set to an And-Relation. An And-Relation also has an "action-group" attribute that points to a set of Source Actions and their outputs, a set of Or-Relations, or a combination of both.

An Or-Relation is satisfied if a path is found to a Source Action that matches any exemplar Source Action in its "action-group" attribute or if any And-Relation in its "action-group" attribute is satisfied. An And-Relation is satisfied if every Or-Relation in its "action-group" attribute is satisfied and a match is found to every Source Action exemplar in its "action-group" attribute.

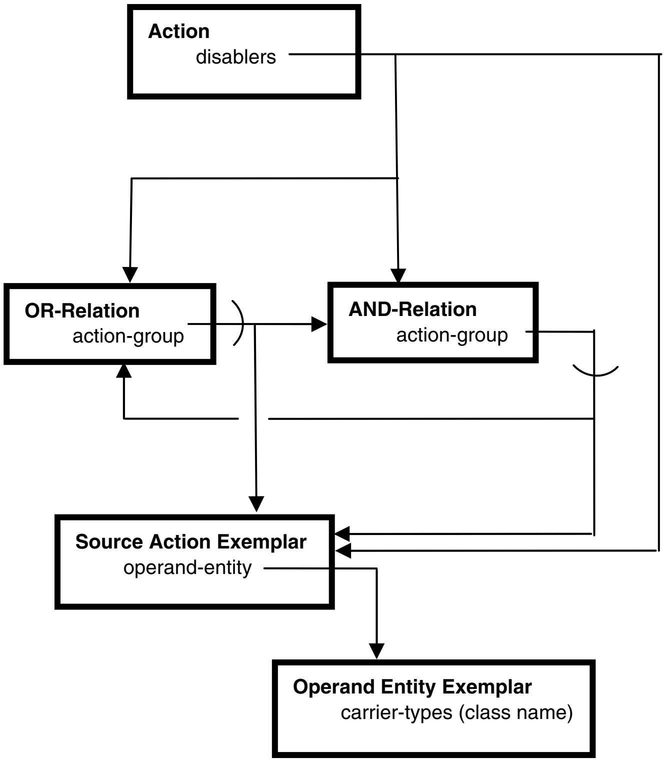

Figure 4 – . Diagram showing relations used to describe a potential hazard to a vulnerable Action.

A hazard specification for a component Action may be an arbitrarily complex And-Or tree composed of alternating layers of And-Relations and Or-Relations. Source Action exemplars are the leaf nodes in any such tree. Figure 5 shows an example of a "disablers" tree that could be constructed according to the scheme represented in Figure 4. Numbers on the left are level numbers. The Or-Relation at Level 1 is the object in the "disablers" attribute of the vulnerable Action. Boxes labeled "Ex" are action exemplars.

A hazard specification for a component Action may be an arbitrarily complex And-Or tree composed of alternating layers of And-Relations and Or-Relations. Source Action exemplars are the leaf nodes in any such tree. Figure 5 shows an example of a "disablers" tree that could be constructed according to the scheme represented in Figure 4. Numbers on the left are level numbers. The Or-Relation at Level 1 is the object in the "disablers" attribute of the vulnerable Action. Boxes labeled "Ex" are action exemplars.

Path Search Procedure

The analysis is based on graph-search to find paths from vulnerable-target components to hazard-source components, using constraints associated with configuration and connections. The prototype currently uses simple depth-first search, but more sophisticated search techniques will be necessary for a fielded application.

The search starts only at the functional actions that are manifested in a component's current mode. No search is necessary if there is no hazard-source component in the system that pairs with the vulnerable-target component. That is, there must be a hazard Source whose current modes manifest an Output Action that produces an Operand Entity to which the Target's functional actions are vulnerable.

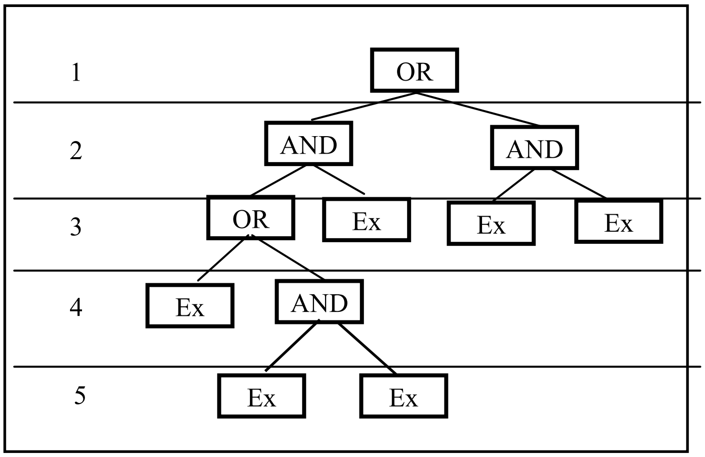

Figure 5 - A hazard specification tree constructed according to the scheme shown in Figure 4.

The path search is further constrained by traversing only components whose current mode manifests an action that is an instance of "Conduction". For example, it makes no sense to traverse a switching element whose current mode is "open." The Conduction Action must also match the "carrier type" of the hazard entity.

Finding a Hazard/Vulnerability Pair

Using the set of constraints on path traversals, we define a hazard in the context of a system configuration. The search compares vulnerability specifications (disabler exemplars) for active functions with actual Source Actions and Entities in the system configuration, and finds propagate paths.

If the current mode of some system component, C_1, manifests an Action, A_1 that is vulnerable to (disabled by) a source Action exemplified by ExS_1, then the search will report any other component, C_2, as the source of a hazard to the vulnerable Component C_1 and Action A_1 if the following conditions are met:

(1) The current mode of Component C_2 manifests a source Action, S_2, which is a member of the class of the (disabling) Source Action Exemplar, ExS_1 of Component $C_{1.}$

(2) The Operand Entity Ent_2 of Action S_2 is a member of the class of the exemplar Entity, $ExEnt_1$, in the Operand-entity attribute of ExS_1 of C_1.

(3) Every Component on the path from Component C_2 to Component C_1 has a "current-mode" manifesting a Conduction Action whose "operand" attribute is of the same class as the problematic Operand Entity Ent_2 output by Action S_2.

(4) Every Component-Connection between components on the path is a member of a subclass of a class listed in the "carrier-types" attribute of the Operand Entity Ent_2.

If, instead of a single exemplar source Action, a Component Action has a "disablers" specification that is an Or-Relation at the top of an And-Or tree, then *any* condition represented by a child node (in the Or-Relation's "action-group") that is manifested in the system makes the Or-Relation proposition true. If the "disablers" specification is an And-Relation, then *all* conditions represented by its child nodes must be manifested in the system for the proposition to be true.

Analysis Example

Returning to the spacecraft model of Figure 1, we relate the elements of the model to the representations we have described. The functional Actions of both transmitters are instances of the Send type of action. The operand of the Send Action is data. The transmitters receive power from the power subsystem, SC1_PwrSpply, through Power connections.

The first step in the hazard analysis is the search for matches to the "disablers" exemplars of the transmitters' Send Actions that are associated with its ON mode. The "disablers" exemplar Actions for the transmitters are instances of source Output Actions with operands that are instances of Electrical-Noise, which may be propagated in either direction through Power connections. The tool finds that when the thermal system, SC1_ThermalSys, is also in its ON mode, it manifests a side effect action of generating electrical noise (i.e. an Output Action with an operand entity of type Electrical-Noise). This type of Action matches the vulnerable transmitter action's "disablers" exemplar Output Action, so a match is found.

The second step in the analysis is to find a path from the thermal system to the transmitter. The only "carrier-type" for Electrical-Noise is Power, so connections of other types are not followed. A path to the primary transmitter is found, and the Hazard Identification Tool reports that a hazard exists: the Output of Electrical-Noise by the thermal system reaching the primary transmitter when the system configuration consists of the transmitter, power supply, and thermal systems in their ON modes.

The paths from the identified source of electrical noise generated by the thermal system to the two transmitters are shown in Figure 6.

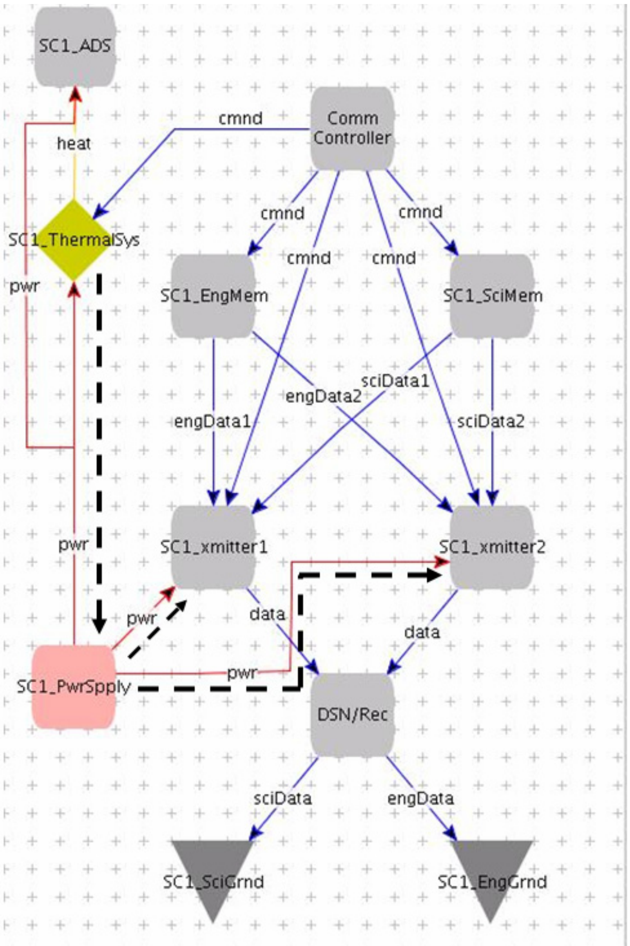

Figure 6 - Model of spacecraft systems from **Figure 1**. Broken line arrows indicate paths from the source of the hazard to the vulnerable components, two radio transmitters.

The redundancy mitigation for any problem with the primary transmitter is to turn it off and use the backup transmitter, SC1_xmitter2. But if the model is transitioned to the state where the primary transmitter is turned off and the secondary transmitter is turned on, the secondary transmitter, too, is found to be vulnerable to noise from the thermal system. A possible resolution to the problem might be to temporarily turn the thermal system off while data is being transmitted. The systems engineer, on further investigation, might simply determine that the actual magnitude of electrical noise from the thermal system as designed does not pose a significant threat to transmission.

If the thermal system designers are not in close communication with the designers of the power and telemetry subsystems, the hazard posed to successful data transmission might go undetected until system integration. This scenario would be unlikely for the highly simplified spacecraft model we used, but similar problems could occur with the complex subsystems of a real spacecraft.

4. Estimating Impacts

For diagnosis and impact assessment, it is useful to consider not only the worst case, but the degree of impact in a particular system configuration. The analysis is based on a way to express the strength (or local severity) of the threat posed by an Entity and the degree of vulnerability of a target Action to exposure to that Entity. The objective of the analysis is to compute the impact of disabling threats as they propagate through a system and finally affect a vulnerable function of a component or subsystem. The approach is to combine the actual strength of a threat in a configuration with the degree of vulnerability.

In the spacecraft example of Figure 6, the impact on the transmitters of electrical noise is generally a slower transmission rate rather than the total loss of the transmission function, so the transmission Action could be assigned a moderate degree of vulnerability to electrical noise. If the thermal system generates a high level of noise in a configuration, we would assign the noise-output Action a high value of (actual) threat strength.

We use the term "weight" to capture degree of "worst case" vulnerability. It can be viewed as a measure of the importance of the threat when it is at maximum strength. When a user assigns a numeric value of weight W_{Ex} to the hazard exemplar (of a Source Action) for a vulnerable Action, the question could be phrased as: "Given that the magnitude of the threat is the worst conceivable for this system (i.e., $E_S = 1$), what would be the impact of the threat, from 0 to 1, on the performance of the vulnerable action?"

We use the term "effectiveness" (also varying from 0 to 1) to capture both the "actual" threat strength in a configuration and the actual performance of a component's intended functional actions. Source Actions can themselves vary in their effectiveness.

The terms "effectiveness" and "weight" are also intended to accommodate future enhancements to analyze "enablers" of functions. "Enablers" are complementary to "disablers". A function may require the supporting Actions of other components ("enablers'). These enablers can offer varying actual degrees of support to a function (i.e., effectiveness) and may have varying degrees of importance to that function (i.e., weight). The analysis will use the same data structures and the same computations to determine the ultimate positive impact of "enablers" that are used to determine the ultimate negative impact of "disablers."

Impact severity is measured by a reduction in effectiveness. The strength of disabling impacts on an action is measured by the extent to which the Action's effectiveness is actually reduced by a hazardous effect in a given system configuration. The maximum "effectiveness" for any Action is 1. A reduction in a vulnerable Action's effectiveness, E_V, by a disabler condition is given by an impact estimate I. The reduced effectiveness is:

$$E_V = 1 - I \qquad (1)$$

Impacts on the effectiveness of vulnerable Actions are computed as a part of the analysis process. Different formulas are used to compute the value of I for exemplar Actions, Or-Relations, and And-Relations.

If a vulnerable Action's "disablers" attribute contains an Action exemplar, *Ex*, for which the path search procedure has found a matching hazard source Action, *S*, the impact function is simply the product of the weight, W_{Ex}, specified for the exemplar Action *Ex* and the effectiveness of the matching source action, E_S , or:

$$I = W_{Ex} E_S \qquad (2)$$

Here, the effectiveness of the Action exemplar is considered to be identical to the effectiveness of the matching source Output Action, with no intervening effects on the propagation path. This formula will be modified when we introduce numeric estimates of path transmission effectiveness in the next section.

If the vulnerable action would be completely disabled by the worst conceivable manifestation of the exemplar Action, then W_{Ex} would be assigned a value of 1, and the vulnerable Action's effectiveness, E_V, would be 0 if a matching source action's effectiveness were indeed 1 in some configuration of the system.

Figure 7 shows how the effectiveness of a vulnerable action is reduced by a source action that matches the vulnerable action's "disablers" exemplar. The exemplar is part of the vulnerable action's definition while the vulnerable action is part of the system model. The link between the source action and the exemplar may be broken by some change in the system's configuration that blocks the path by which the entity it outputs reaches the vulnerable action. The source action is shown to have an effectiveness of 0.9, indicating that its effectiveness is also slightly reduced from the maximum of 1.0 because, in this configuration, it is also exposed to some source action to which it is vulnerable.

C_E and E_S are allowed to range in value from 0 to 1 so that the values of I produced by Equation (2) and the values of E_V returned by Equation (1) also range from 0 to 1. A value of I close to 1 indicates that the hazard should be considered to be very severe, reducing E_V to a value near 0. At the other extreme, a value of I close to zero indicates that the hazard should be considered to have only a minor impact on the effectiveness of Action V, and E_V will be close to 1.

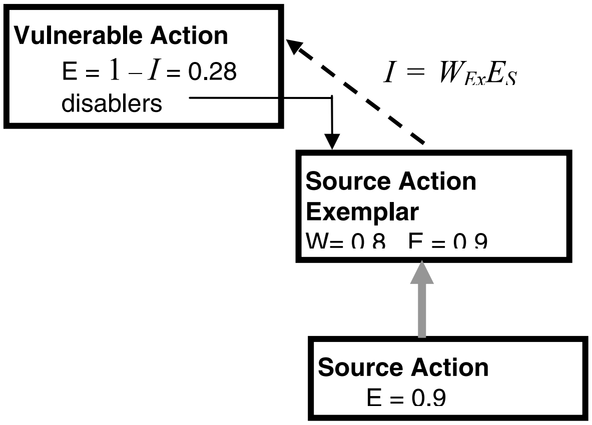

Figure 7 – Diagram showing the relationship of a vulnerable action, an exemplar of the kind of source action to which it is vulnerable, and a matching action found in the model configuration.

In a "disablers" tree structure such the one illustrated in Figure 4, every And-Relation and Or-Relation in the tree structure at level L has an impact on its parent node at level L-1 expressed by the formula:

$$I_{L-1} = W_L E_L\left(I_{1,L+1}, \cdots, I_{n,L+1}\right) \qquad (3)$$

Where $E_L(I_{1,L+1}, \ldots I_{n,L+1})$ is the effectiveness of an And or Or-Relation as a function of the impacts of its n child nodes at level L+1 in the tree structure.

We define the formula for the effectiveness of an Or-Relation as follows. Assume that the Or-Relation has n child nodes in its "action-group" attribute and the i$^{\text{th}}$ child node has a weight of W_i and an effectiveness of E_i. The effectiveness of the Or-Relation thus is:

$$E_{OR} = \max_{i=1}^{n}\left(W_i E_i\right) \qquad (4)$$

Intuitively, it might seem that the effectiveness of an And-Relation at any position in an And-Or tree should be the product of the impacts of all of its child nodes (the Actions and Or-Relations in its "action-group"), or:

$$E_{AND} = \Pi_{i=1}^{n}\left(W_i E_i\right)$$

However, the simple formula produces results that are the opposite of what weights should indicate. For example, if a child node is assigned a weight near zero, the impact of the And-Relation would always be close to zero regardless of the weight and effectiveness of other child nodes. A more complex formula for And-Relation effectiveness of works:

$$E_{AND} = \Pi_{i=1}^{n}\left[1 - W_i\left(1 - E_i\right)\right] - \Pi_{i=1}^{n}\left(1 - W_i\right) \qquad (5)$$

This formula handles "weight" as a measure of importance with the meaning that would reasonably be associated with an And-Relation's effectiveness.

- If one child node were assigned a weight near zero, it would contribute very little to the value of E_{AND}.
- If all child nodes but one were to be assigned a weight near 0, then the effectiveness of the And-Relation would be approximately equal to the product of the weight and effectiveness of the single child node with the nonzero weight. (The condition represented by that child node would be the only one of real importance, so the And-Relation's effectiveness would have the same value.)
- If a child node has a weight of 1 and an effectiveness of 0 then the effectiveness of the And-Relation is 0. (Even though the condition is critically important, it is totally ineffective and therefore passes this ineffectiveness to the And-Relation as a 0 impact.)
- If all child nodes have a weight of 1 and an effectiveness of either 0 or 1, the formula must return either a 0 or a 1, thus reducing to the Boolean And operator. (Boolean values may be preferred early in design and can be replaced with estimates as more information is acquired.)

The diagram in Figure 8 illustrates how the effectiveness is computed for an Action with an And-Or tree of "disablers". The "effectiveness" of each And-Relation and Or-Relation is set to the value of the collective impact of its child nodes. Equation (4) is used for computing the effectiveness of all Or-Relations and Equation (5) is used for all And-Relations. If the "disablers" attribute of the vulnerable Action contains a single Action exemplar rather than a tree structure, Equation (2) is used to compute the impact of the exemplar Action matched to a manifested Action and Equation (1) is used to compute the effectiveness of the vulnerable Action.

5. Propagation Effectiveness

The existence of a path between two components is essentially a Boolean proposition; either one or more paths exist or none exist. Conducting Actions can be assigned effectiveness ratings. The Conducting Actions of components on a path are similar to the conditions of an And-relation, and can be used to compute propagation effectiveness. For example, on the path for electrical noise from the spacecraft's thermal system to the transmitters in Figure 6, the power supply (SC1_PwrSpply) is specified to be a potential conductor of electrical noise. If a switch was on the same path, both the switch and the power supply could be viewed as two child nodes of an And-Relation. Both would have to be in a mode in which electrical noise is conducted (the ON mode for the power supply and the CLOSED mode for the switch). The switch and the power supply could have different effectiveness ratings for conducting noise, each between 1 and 0.

After assigning numerical values to the "effectiveness" attributes of paths and Conducting Actions, the effectiveness of an exemplar Action is no longer identical to the effectiveness of the matching source action. Given that there are *n* paths and the effectiveness of the i^{th} path is E_i, the effectiveness of an Action matched with an Action exemplar and modified by the transmission effectiveness is:

$$E_{Ex} = E_S \max_{i=1}^{n} \left(E_i \right) \tag{6}$$

The effectiveness of the i^{th} path with *m* components between the source component and the vulnerable component is:

$$E_i = \Pi_{j=0}^{m} \left(E_j \right) \tag{7}$$

Every Conducting Action on a path can be assigned its own "disablers" specification. The measure of Conducting Action effectiveness enables search for possible mitigations involving reduction of transmission effectiveness of the hazardous entity (e.g., filters).

The complete rating scheme, if applied to the simple spacecraft model of Figure 6, could express the hazard posed by the electrical noise generated by the thermal system to the transmitters as a function of:

- Effectiveness of the noise generation
- Effectiveness of noise propagation by components on the path between the thermal system and transmitters
- Weight (worst case impact) of the potential noise transmission hazard on successful data transmission from the spacecraft to ground

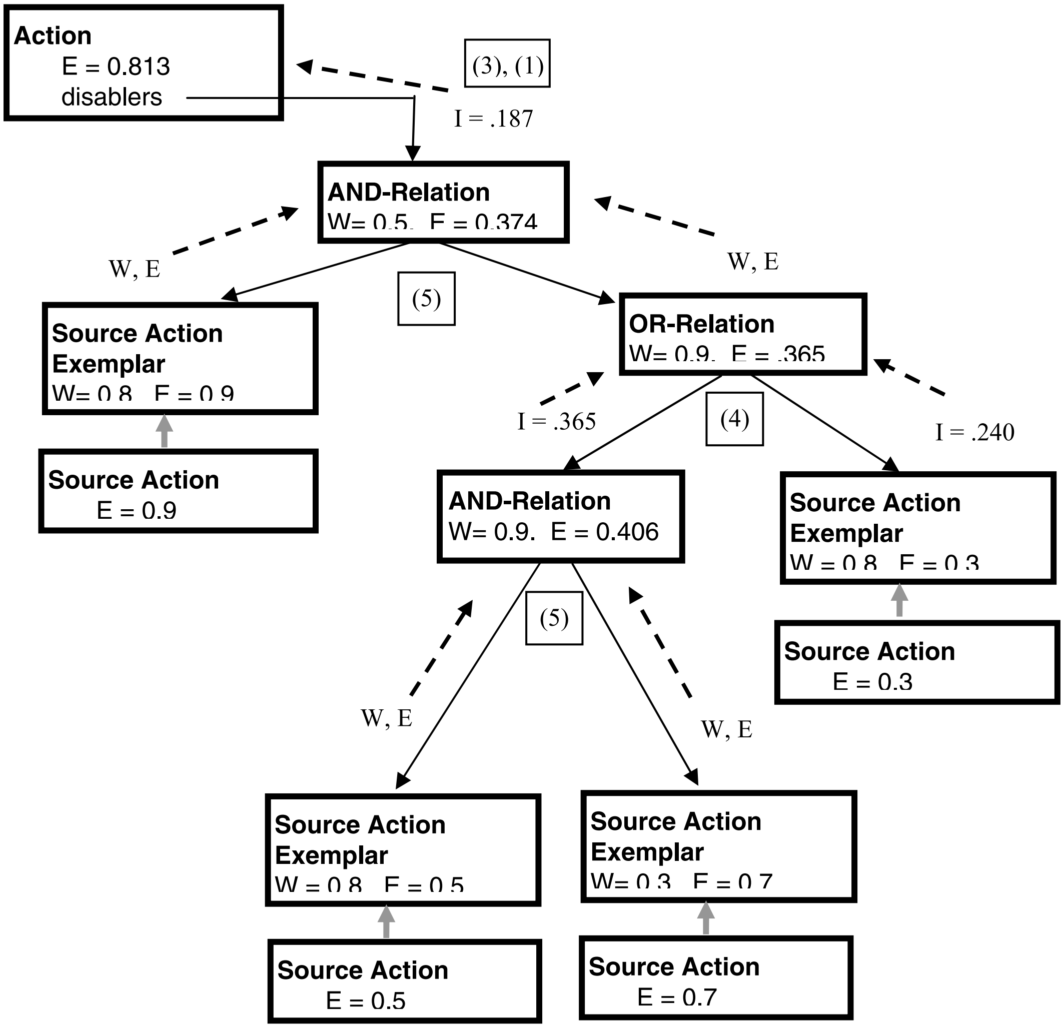

Figure 8 - Example Action having a "disablers" And-Or tree. The solid black arrows represent the relations. The dashed arrows show the direction of data flows. The gray arrows indicate a match of a source action exemplar to a source action manifested in the system model. Numbers in parentheses indicate the equations that determine values at the next level.

The output report of an analysis would be a listing of functional actions whose effectiveness has been reduced below some user-chosen threshold, generally close to zero. The fractional values of effectiveness computed during the analysis could be converted to the severity ratings used in five-by-five risk rating matrices used by NASA. Table 1 shows an example of mapping effectiveness to severity.

Table 1. Mapping Action Effectiveness to Hazard Rating

Effectiveness Range	Severity Rating
0 to 0.2	Very High
0.2 to 0.4	High
0.4 to 0.6	Moderate
0.6 to 0.8	Low
0.8 to 1.0	Very Low

6. KNOWLEDGE ACQUISITION

The Hazard Identification Tool, unlike other systems engineering tools, analyzes hazard impacts on vulnerable functions by using a connectivity model of propagation paths in a system. In HIT, functions and "physical" models are naturally linked. To extract function, risk and connectivity data from requirements text, we have developed a text parsing and matching approach, using our ontology and mapping words [12, 17]. This type of extracted electronic data can be reused and evolved, and thus supports traceability and consistency.

It can be a challenge to get needed information on components, hazards, vulnerabilities. The approach we have taken is a version of the common one used in advanced risk analysis tools. We use libraries of templates, archived knowledge, and standard nomenclature. The libraries of critical entities, vulnerabilities, hazards and risks make it possible to apply the equivalent of hazard checklists to a system connectivity model. Such libraries can be extended as needed and reused in later phases or other projects.

Definitions for each class of component or subsystem in the design model would be stored in the libraries. Each type of component or subsystem would have one or more operational Modes. One or more Actions would be associated with each Mode. These Actions could be component functions or other behaviors (e.g., side effects) that can be manifested in the mode. Generally, in the current analysis approach, these actions generate or transmit Entities. In the spacecraft example in Figure 1, these entities are power, electrical noise, heat, and data.

For each target function in the library, vulnerabilities would be specified as one or more types of disablers. Each disabler would have an associated estimate of worst-case impact, which indicates the vulnerability of the function to that type of disabler. Likewise, for each hazard in the library, types of disablers of the effectiveness of the hazard-producing action would be specified. The default "estimates" of importance would be Boolean 1's or 0's. Users could refine and extend disablers and strength estimates.

In the spacecraft example, it would be the responsibility of the transmitter design group to identify electrical noise as a potential problem for data transmission to earth. Initially, this group might be reluctant to assign worst-case weight of this disabler of less than 1. This vulnerability information could alert the thermal system design group to look within their subsystem for electrical noise sources.

In most cases, it would be best to screen first for potential pairs and paths with the Boolean approach. As the design progresses and as the analysis highlights potential problems, the estimates could be refined to values varying from 0 to 1 on subsequent analysis iterations.

8. CONCLUSIONS AND RECOMMENDATIONS

Our approach to system risk analysis uses early design information to improve the detection and assessment of hazards and risks that involve propagation and complex interactions across subsystems. Identifying hazard-vulnerability pairs and paths across subsystems can help engineers to more selectively yet more completely evaluate the integrated impact of risks. Automating and aiding the process of hazard analysis can both save time and help detect some severe cross-subsystem hazards that might escape attention. It can also indicate ways to control risk.

Search for propagation paths and estimates of "effectiveness" can start simply and can be incrementally elaborated as design progresses. This approach should be applicable to hazard analysis, fault tree analysis and FMECA. It would be interesting to extend this work to validation of fault trees and FMECAs.

When vulnerabilities are specified with a disabler tree, analyses can mix logical combinations with search through dynamic configurations associated with procedures. In our prototype, the current mode of each component in a system is set when the model is initialized. In future work, operational scripts could be evaluated. Procedure models would be linked to controllable modes of the components in system models and could command a series of mode transitions. The current modes of components would be changed according to the system's operational procedures (and possibly failures). After each configuration change, estimates would be made of hazard strength for actions in each Source component's new mode and impacts on the effectiveness of vulnerable functions.

Human subsystems can both contribute to and be impacted by functional failures, hazards and risks. Current hazard analysis methods typically have difficulty analyzing these interactions. Our approach can use HIT ontologies and models include to human subsystems in the analysis.

Thus far, we developed prototypes and demonstrated them on small test cases. Application awaits resources to develop a mature prototype. We have developed a concept for extending Preliminary Hazard Analysis beyond worst case impacts, using vulnerability ratings. We are also working on concepts for reuse of the models and analysis methods in fault diagnosis and anomaly response in operations. The following additions to the analysis methods and models would extend reusability.

- Extend the disabler tree beyond vulnerable functions to address vulnerability of entities (e.g., injury or spoilage due to a toxic chemical).
- Extend the capability for estimating impacts by including failures to receive a desirable or necessary entity or entity attribute value (e.g., chemical reactor deprived of needed catalyst).
- Flesh out the use of the disabler concept for hazard controls and mitigations. These controls and mitigations would be disablers of hazards production or propagation paths to the vulnerable entity. Thus, hazard analysis methods can be used to analyze effectiveness of mitigations.
- Identify mitigation strategies by searching for system configurations that would decrease the effect of a threat on a function. (An action that outputs a hazardous substance may have its own vulnerabilities that might be exploited to mitigate the hazard.) This capability may be useful for identifying hazard controls during design and for anomaly response during operations.
- Extend the path analysis so that chains of functional dependencies can be inferred in the same way as chains of hazard impacts.
- Extend the impact estimation scheme to express the effectiveness of an entity source at providing the entity with specified attribute values (e.g., heating a fluid to a specified temperature). Similarly, extend the rating scheme to numerically express the vulnerability of a component to the value of a specified entity attribute (e.g., entity that is too hot for processing).
- Extend the impact analysis to take account of probabilities of the occurrences of actions and propagations. This extension would be useful for evaluating the importance of hazards and for deriving and validating hazard likelihood ratings. HIT would then be able to compute both likelihood ratings and severity ratings for a NASA risk matrix.

REFERENCES

[1] K. Allenby and T. Kelly, “Deriving Safety Requirements Using Scenarios.” *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, August 2001, 228- 235.

[2] Center for Chemical Process Safety, *Layer of Protection Analysis, Simplified Process Risk Assessment*, American Institute of Chemical Engineers, New York, NY, 2001.

[3] S. L. Cornford, M. S. Feather, and K. A. Hicks, “DDP – A Tool for Life-cycle Risk Management,” *2001 IEEE Aerospace Conference Proceedings*, March 2001.

[4] S. A. Henning and R Paasch, “Diagnostic Analysis for Mechanical Systems,” *Proceedings of DETC '00*, 2000 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Baltimore, MD, September 2000.

[5] E. Hollnagel, *Accident Analysis and Barrier Functions*. Halden, Norway: Institute for Energy Technology, 1999.

[6] S. Kmenta and K. Ishii, “Advanced FMEA Using Meta Behavior Modeling for Concurrent Design of Products and Controls,“ *Proceedings of DETC ’98*, 1998 ASME Design Engineering Technical Conferences, Atlanta, GA, September 1998.

[7] Nancy Leveson, “A New Accident Model for Engineering Safer Systems.” *Safety Science*, 42:4, April 2004.

[8] J. T. Malin, L. Fleming and D. R. Throop, “Predicting System Accidents with Model Analysis during Hybrid Simulation,” *Proceedings of Business and Industry Symposium, Advanced Simulation Technologies Conference*, Simulation Councils, Inc., 2002, 155-160.

[9] J. T. Malin, L. Fleming and D. R. Throop, “Hybrid Modeling for Scenario-Based Evaluation of Failure Effects in Advanced Hardware-Software Designs,” *Model-Based Validation of Intelligence*, Technical Report SS-01-04, AAAI Press, Menlo Park, CA, 2001.

[10] J. T. Malin and D. L. Schreckenghost, “Problems of Mapping between Functions and Device Phenomena.” *Working Notes, AAAI-96 Workshop: Modeling and Reasoning with Function,* Portland, OR, 1996: 14-15.

[11] J. T. Malin, D. R. Throop, L. Fleming and L. Flores, “Computer-Aided Identification of System Vulnerabilities and Safeguards during Conceptual Design,” *2004 IEEE Aerospace Conference Proceedings,* March 2004.

[12] J. T. Malin, D. R. Throop, L. Fleming and L. Flores, “Transforming Functional Requirements and Risk Information into Models for Analysis and Simulation,” *2005 IEEE Aerospace Conference Proceedings*, March 2005.

[13] NASA Exploration Systems Mission Directorate. *Methodology for Conduct of Project Constellation Hazard Analyses*. (Document 0000028585, 2005).

[14] Robert B. Stone, I. Y Tumer and M. Van Wie. “The Function-Failure Design Method,” *Journal of Mechanical Design*, **127**: 3 (2005), 397-407.

[15] Jaffee Suardin. “*The Integration of Dow’s Fire and Explosion Index into Process Design and Optimization to Achieve an Inherently Safer Design*.” Texas A & M Chemical Engineering Masters Thesis, August 2005.

[16] Jerome Tixier, G. Dusserre, O. Salvi and D. Gaston. “Review of 62 Risk Analysis Methodologies of Industrial Plants,” *Journal of Loss Prevention in the Process Industries*, **15** (2002), 291-303.

[17] D. Throop, “Reconciler: Matching Terse English Phrases,” *Proceedings of 2004 Virtual Iron Bird Workshop*, NASA Ames Research Center, April, 2004.

[18] D. R. Throop, J. T. Malin and L. Fleming. 2001. “Automated Incremental Design FMEA,” *2001 IEEE Aerospace Conference Proceedings,* March 2001.

[19] V. Venkatasubramanian, Jinsong Zhao and Shankar Viswanathan. “Intelligent Systems for HAZOP Analysis of Complex Process Plants,” *Computers and Chemical Engineering*, **24** (2000), 2291-2302.

[20] P. J. Wilkinson and T. P. Kelly. “Functional Hazard Analysis for Highly Integrated Aerospace Systems,” *Proceedings of IEE Seminar on Certification of Ground/Air Systems*, 1998.

ACKNOWLEDGEMENTS

This work has been funded by the NASA Exploration Systems Mission Directorate, a legacy project from the NASA Engineering for Complex Systems program. Thanks to David Throop, Luis Flores, Dennis Lawler and Carroll Thronesbery for working with us to develop the overall concept and context for this analysis work.

BIOGRAPHY

Jane T. Malin *is Senior Technical Assistant in the Intelligent Systems Branch, Automation, Robotics and Simulation Division, Engineering Directorate, NASA Johnson Space Center, where she has led intelligent systems research and development since 1984. Her work includes the CONFIG hybrid simulation tool, intelligent user interface and intelligent agents for control of space systems, and teamwork tools for anomaly response teams. Her 1973 Ph.D. in Experimental Psychology is from the University of Michigan.*

Land D. Fleming *is a Computer Systems Specialist supporting the NASA Johnson Space Center Automation, Robotics, and Simulation Division since 1990. He has been involved in both the development of computer simulation tools and their application to space systems. His 1987 M. S. in Computer Science is from De Paul University.*